COMPUTER ASSIGMENT

INVERSE PROBLEMS ARE AMONG THE MOST CHALLENGING COMPUTATIONS IN SCIENCE AND ENGINEERING BECAUSE THEY INVOLVE DETERMINING THE PARAMETERS OF A SYSTEM THAT is only observed indirectly. For example, we might have a spectrum and want to determine the species that produced it as well as their relative proportions. Or we may have sonar measurements of a containment tank and want to know whether it has an internal crack.

Here is this issue's homework assignment: given a blurred image and a linear model for the blurring, reconstruct the original image. This linear inverse problem illustrates the impact of *ill-conditioning* on the choice of algorithms.

Ill-Conditioning

Consider a linear system of equations

$$K\mathbf{f} = \mathbf{g}$$

Tools

The major tool used in this project is the singular value decomposition of a matrix.[2] Any real matrix A of dimension $m \times n$ (with $m \geq n$) has a representation as

$$A = U\Sigma V^T$$

where $U^TU = I$, $V^TV = I$, and Σ has nonnegative entries σ_i ($i = 1, \ldots, n$) on its main diagonal and zeros elsewhere.

The matrix U is $m \times m$, V is $n \times n$, and Σ is $m \times n$. The singular values σ_i are the square roots of the eigenvalues of A^TA, and the columns of V are the eigenvectors of that matrix. The columns of U are the eigenvectors of AA^T. Computation of the singular value decomposition is more stable than forming A^TA and computing the eigendecomposition.

where K is an $n \times n$ matrix, and $\mathbf{f}$ and $\mathbf{g}$ are vectors. Suppose K is scaled so that its largest singular value is $\sigma_1 = 1$. If the smallest singular value is $\sigma_n \approx 0$, then K is *ill-conditioned*. We distinguish two types of ill-conditioning:

- The matrix K is considered numerically rank deficient if there is a j such that $\sigma_j \gg \sigma_{j+1} \approx \ldots \approx \sigma_n \approx 0$. That is, there is an obvious gap between large and small singular values.
- If the singular values decay to zero with no particular gap in the spectrum, we say the linear system $K\mathbf{f} = \mathbf{g}$ is a *discrete ill-posed problem*.

Computing accurate approximate solutions of discrete ill-posed problems is extremely difficult, especially because in most real applications, $\mathbf{g}$ is not known exactly. Rather, the collected data typically has the form

$$\mathbf{g} = K\mathbf{f} + \eta,$$

where η is a vector representing (unknown) noise or measurement errors. The goal, then, is given an ill-conditioned matrix K and a vector $\mathbf{g}$, compute an approximation of the unknown vector $\mathbf{f}$.

Naïvely solving $K\mathbf{f} = \mathbf{g}$ usually does not work because the matrix K is so ill-conditioned. Instead, *regularization* is used to make the problem less sensitive to the noise.

Tikhonov Regularization

The best-known regularization procedure—*Tikhonov regularization*—computes a solution of the damped least-squares problem:

$$\min_{\mathbf{f}} \left\{ \|\mathbf{g} - K\mathbf{f}\|_2^2 + \alpha^2 \|\mathbf{f}\|_2^2 \right\}. \tag{1}$$

The extra term $\alpha^2\|\mathbf{f}\|_2^2$ imposes a penalty for making the norm of the solution too big, which reduces the effect of small singular values. This regularized problem is also a least-squares problem.

Problem 1. Show that Equation 1 is equivalent to the linear least-squares problem,

$$\min_{f}\left\|\begin{bmatrix}\mathbf{g}\\0\end{bmatrix}-\begin{bmatrix}K\\\alpha I\end{bmatrix}\mathbf{f}\right\|_2^2. \qquad (2)$$

The scalar α (called a *regularization parameter*) controls the solution's degree of smoothness. Note that $\alpha = 0$ implies no regularization; the computed solution to Equation 2 with $\alpha = 0$ will likely be horribly corrupted with noise. On the other hand, if α is large, then the computed solution cannot be a good approximation of the exact f. Choosing an appropriate value for α is not a trivial matter. Various algorithms appear elsewhere in the literature,[1] but we use a manual approach here.

Let's turn to the problem of solving the least-squares problem encountered in Equation 2 in Problem 1.

Problem 2. Show that if K has a singular value decomposition $K = U\Sigma V^T$, then Equation 2 can be transformed into the equivalent least-squares problem,

$$\min_{f}\left\|\begin{bmatrix}\hat{\mathbf{g}}\\0\end{bmatrix}-\begin{bmatrix}\Sigma\\\alpha I\end{bmatrix}\hat{\mathbf{f}}\right\|_2^2, \qquad (3)$$

where $\hat{\mathbf{f}} = V^T\mathbf{f}$ and $\hat{\mathbf{g}} = U^T\mathbf{g}$.

Problem 3. Determine a formula for the solution to Equation 3. Hint: you should set the derivative of the minimization function to zero and solve for $\hat{\mathbf{f}}$.

This gives us an algorithm to determine the Tikhonov solution to a discrete ill-posed problem.

Truncated Singular Value Decomposition

Another way of regularizing the problem is to truncate the singular value decomposition (SVD). Problem 4 demonstrates how to express the solution to the least-squares problem in terms of the SVD.

We can see that trouble occurs in $\mathbf{f}_{LS}$ if a small value of σ_i divides a term $\mathbf{u}_i^T\mathbf{g}$ that is dominated by error. In such cases, $\mathbf{f}_{LS}$ will be dominated by error.

Problem 4. Show that the solution to the problem

$$\min_{f}\left\{\|\mathbf{g}-K\mathbf{f}\|_2^2\right\}$$

is

$$\mathbf{f}_{LS} = V\Sigma^{+}U^T\mathbf{g} = \sum_{i=1}^{n}\frac{\mathbf{u}_i^T\mathbf{g}}{\sigma_i}\mathbf{v}_i,$$

where $\mathbf{u}_i$ is the ith column of U, and $\mathbf{v}_i$ is the ith column of V.

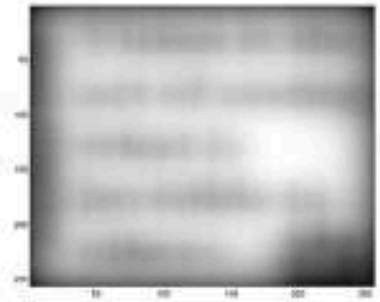

Figure 1. Use two algorithms to read the text in this blurred message.

To overcome this, Richard Hanson (as well as James Varah) suggested truncating the previously mentioned expansion,[3,4]

$$\mathbf{f}_t = \sum_{i=1}^{p} \frac{\mathbf{u}_i^T \mathbf{g}}{\sigma_i} \mathbf{v}_i,$$

for some value of $p < n$.

Now we have all the tools in place to solve a deblurring problem in image processing. Suppose we have a blurred, noisy image G (along with some knowledge of the blurring operator), and we want to reconstruct the true original image F. This is an example of a discrete ill-posed problem, in which the vectors in the linear system $\mathbf{g} = K\mathbf{f} + \eta$ represent the image arrays stacked by columns to form vectors. In Matlab notation, it looks like this:

- $\mathbf{f}$ = reshape(F, n, 1),
- $\mathbf{g}$ = reshape(G, n, 1).

The goal in this problem is given K and G, reconstruct an approximation of the unknown image F.

If we assume F and G contain $\sqrt{n} \times \sqrt{n}$ pixels, then $\mathbf{f}$ and $\mathbf{g}$ are vectors of length n, and K is an $n \times n$ matrix representing the blurring operation. In general, this matrix is too large to use the SVD. However, in some cases, we can write K as a Kronecker product, $K = A \otimes B$, and then we can use the SVD.

A Few Facts on Kronecker Products[5]

The Kronecker product $A \otimes B$, in which A is an $m \times m$ matrix, is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ a_{21}B & a_{22}B & \dots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mm}B \end{bmatrix}.$$

Theorem 1. If $A = U_A \Sigma_A V_A^T$ and $B = U_B \Sigma_B V_B^T$, then $K = U \Sigma V^T$, where $U = U_A \otimes U_B$, $\Sigma = \Sigma_A \otimes \Sigma_B$ and $V = V_A \otimes V_B$.

Therefore, computing the SVD of a large matrix is possible if it is the Kronecker product of two smaller ones. On the Web page for this column (http://computer.org/cise/homework/v5n3.htm), there is a sample Matlab program, `projdemo.m`, illustrating this property.

To solve our image-deblurring problem, we must operate carefully with the small matrices; otherwise, storage quickly becomes an issue. Again, see the sample program for guidance. With the Kronecker product as a tool, we are ready to compute.

<u>Problem 5</u>: Write a program that takes matrices A, B and image G and computes approximations to image F using Tikhonov regularization and Truncated SVD. For each of these two algorithms, experiment to find the valueof the regularization parameter (a for Tikhonov or p for TSVD) that gives the clearest image. In the file of the project you will find the necessary matrices.